

ANT COLONY OPTIMIZATION

WIDHAPRASA EKAMATRA WALIPRANA - 13508080

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: w3w_stay@yahoo.com

ABSTRAK

The Ant Colony Optimization (ACO) atau yang lebih dikenal sebagai algoritma semut merupakan teknik probalistik untuk memecahkan masalah perhitungan dengan menemukan jalur terbaik melalui graf. Algoritma ini terinspirasi dari perilaku semut bersama dengan koloninya dalam mencari makanannya.

Pada dunia nyata, semut berjalan secara random dalam memilih jalurnya pada saat mencari makanan. Setelah menemukan makanan tersebut, semut tadi kembali ke koloninya atau sarangnya dengan meninggalkan suatu pheromone sebagai jejak. Saat semut lain mencari makanan dan menemukan jalur dengan pheromone tersebut mereka tidak cenderung untuk bepergian secara random, tetapi akan mengikuti jejak tersebut dan akan menghasilkan pheromone baru jika pada akhirnya menemukan makanan.

Algoritma semut ini merupakan metode untuk melakukan suatu optimasi kerja. Algoritma semut ini dapat diimplementasikan dalam kehidupan sehari-hari seperti pada kasus Travelling Salesman Problem (TSP), Job-shop Scheduling Problem (JSP), dan kasus lain sebagainya.

Kata kunci: Ant Colony Optimization, Algoritma Semut, Graf, Optimasi

1. PENDAHULUAN

Pernahkah melihat semut berjalan bergerombolan dalam mencari makanan. Gerombolan semut berjalan secara teratur, ada yang ke arah makanan ada pula yang ke arah sebaliknya. Awalnya semut mencari makanan secara random hanya memanfaatkan indra penciumannya yang terkenal tajam, karena semut tidak memiliki indra penglihatan. Namun setelah menemukan makanan, semut akan kembali ke sarangnya dengan meninggalkan suatu pheromone yang berperan sebagai jejak agar semut lain dapat mencium pheromone tersebut sehingga tidak akan mencari makanan secara random lagi. Ini pula alasan ada

semut yang berjalan ke arah sebaliknya dari arah makanan.

Dengan inspirasi perilaku semut tersebut maka diciptakanlah suatu algoritma yang berperan sebagai optimasi dalam mencari jalan terbaik pada suatu graf. Oleh karena itu algoritma ini dinamakan Ant Colony Optimization (ACO).

2. LANDASAN TEORI

Pada bagian ini akan dijelaskan mengenai optimasi, graf yang mengacu pada algoritma semut dan algoritma semut itu sendiri secara lebih detail.

2.1. Optimasi

Optimasi ialah suatu proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dapat dicapai). Dalam bidang matematika optimasi, merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi. Untuk dapat mencapai nilai optimal baik minimal atau maksimal tersebut, secara sistematis dilakukan pemilihan nilai variabel integer atau nyata yang akan memberikan solusi optimal.

Adapun yang dikenal dengan nilai optimal yaitu nilai yang didapat dengan melalui suatu proses dan dianggap menjadi suatu solusi jawaban yang paling baik dari semua solusi yang ada. Nilai optimal dapat dicari dengan dua cara:

- Cara pertama yaitu cara konvensional, yaitu mencoba semua kemungkinan yang ada dengan mencatat nilai yang didapat. Cara ini kurang efektif, karena akan membutuhkan banyak waktu dalam pengecekan sehingga akan menjadi tidak optimal.
- Cara kedua adalah dengan menggunakan suatu formula ataupun media seperti gambar atau grafik sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat.

Contoh persoalan yang memerlukan optimisasi diantaranya adalah:

- Menentukan lintasan terpendek dari suatu tempat ke tempat yang lain.
- Menentukan jumlah pekerja seminimal mungkin dalam melakukan suatu proses produksi dengan

pengeluaran biaya pekerja dapat seminimal mungkin dan hasil produksi semaksimal mungkin.

- Mengatur jalur kendaraan umum agar dapat mencapai semua lokasi.
- Mengatur routing jaringan kabel telepon agar biaya pemasangan kabel tidak terlalu besar dan penggunaannya tidak boros.
- Dan juga contoh-contoh lainnya.

2.2. Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.

Definisi Graf

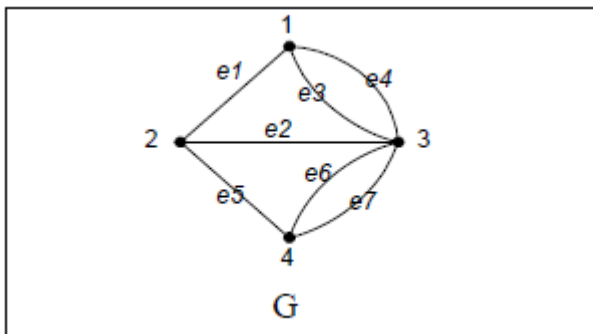
Graf $G = (V, E)$, yang dalam hal ini:

V = himpunan tidak-kosong dari simpul-simpul (*vertices*)

= $\{v_1, v_2, \dots, v_n\}$

E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul

= $\{e_1, e_2, \dots, e_n\}$



Gambar 1. Contoh Graf

Keterangan Gambar 1:

G adalah graf dengan

$V = \{1, 2, 3, 4\}$

$E = \{(1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4)\}$

= $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

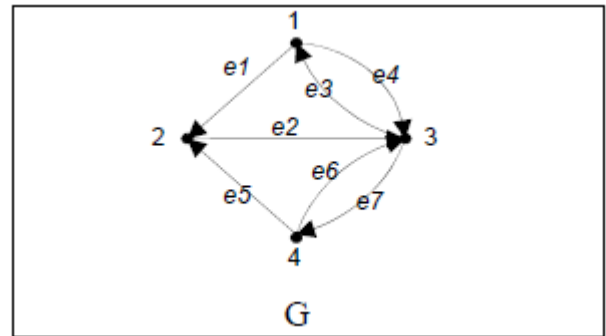
Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

1. Graf tak-berarah (*undirected graph*)

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.

2. Graf berarah (*directed graph* atau *digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

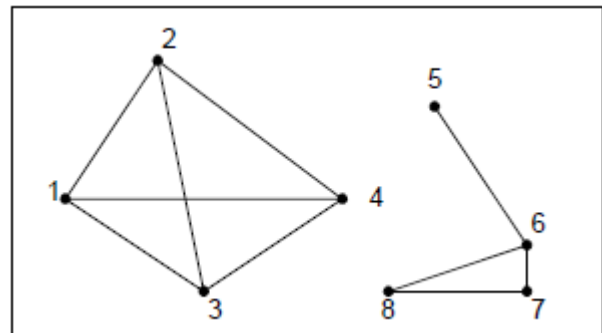


Gambar 2. Contoh Graf Berarah

Dua buah simpul v_1 dan simpul v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 .

G disebut graf terhubung (*connected graph*) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j .

Jika tidak, maka G disebut graf tak-terhubung (*disconnected graph*).



Gambar 3. Contoh Graf Tak-terhubung

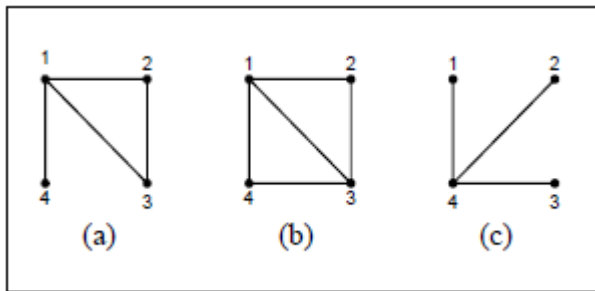
Graf berarah G dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari G diperoleh dengan menghilangkan arahnya).

Graf Hamilton

Lintasan Hamilton ialah lintasan yang melalui tiap simpul di dalam graf tepat satu kali.

Sirkuit Hamilton ialah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.

Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi- Hamilton.



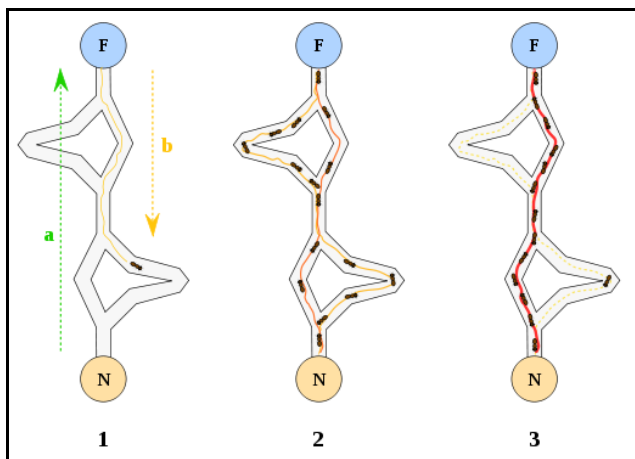
Gambar 4. Graf Hamilton

Keterangan Gambar 4:

- (a) graf yang memiliki lintasan Hamilton (misal: 3, 2, 1, 4)
- (b) graf yang memiliki lintasan Hamilton (1, 2, 3, 4, 1)
- (c) graf yang tidak memiliki lintasan maupun sirkuit Hamilton

2.3. Algoritma Semut

Algoritma semut diperkenalkan oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo.



Gambar 5. Jalur Makanan Semut

Gambar di atas merupakan jalur makanan semut dengan F adalah food source yang berarti sumber makanan dan N adalah nest yang berarti sarang semut.

Gagasan awal algoritma semut ini berasal dari hasil pengamatan sumber makanan dari semut itu sendiri dimana semut secara individual memiliki kemampuan kognitif dalam menemukan jalur terpendek antara sumber makanan dan sarangnya. Berikut adalah kronologis bagaimana semut mencari makanannya:

1. Semut pertama mencari sumber makanan melalui jalur manapun dalam hal ini adalah (a). Kemudian kembali ke sarang (N) dengan meninggalkan jejak pheromone (b).
2. Semut tanpa pandang bulu akan mengikuti empat kemungkinan, tapi dengan pheromone yang lebih kuat, semut lebih tertarik memilih jalur tersebut sebagai jalur terpendek.
3. Akhirnya semut akan mengambil jalur terpendek, dimana jalur lainnya akan semakin kehilangan pheromone yang telah menguap.

Dalam berbagai percobaan pada koloni semut yang diberikan dua pilihan jalur menuju sumber makanan yang tidak sama panjang, koloni semut cenderung memilih jalur terpendek. Sebuah model yang menjelaskan perilaku semut tersebut akan dijelaskan berikut:

1. Seekor semut (sebut saja “blitz”) kurang lebih berjalan atau berkeliling di sekitar atau tidak jauh dari koloninya.
2. Jika menemukan sumber makanan, ia langsung akan kembali ke sarangnya dan meninggalkan jejak pheromone
3. Dengan pheromonenya, semut terdekat kurang lebih akan langsung mengikuti jalur yang telah dilewati blitz.
4. Setelah kembali ke koloninya semut ini akan semakin memperkuat jalur dengan pheromone yang ia hasilkan.
5. Jika terdapat dua rute yang mungkin untuk mencapai sumber makanan yang sama, jalur terpendek akan lebih banyak dipilih oleh semut dibandingkan yang lebih panjang.
6. Jalur terpendek akan semakin banyak peminatnya dan akan lebih menarik dikarenakan semakin banyak pheromone yang dihasilkan.
7. Jalur yang panjang akan ditinggalkan dan akhirnya pheromone yang terdapat di sana akan menguap dan hilang.
8. Akhirnya semua semut telah mendapatkan pemikiran untuk memilih jalur terpendek akibat ulah blitz si semut pertama tadi.

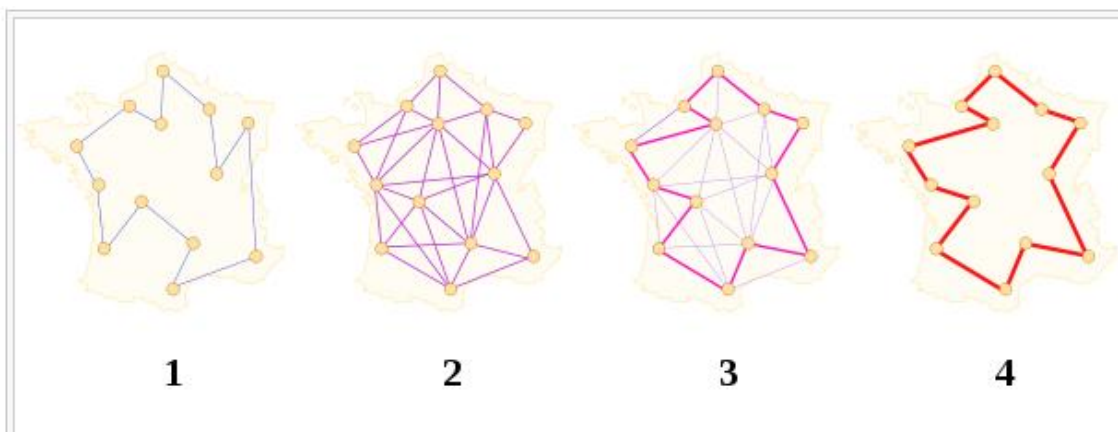
Semut menggunakan lingkungannya sebagai media komunikasi. Mereka bertukar informasi secara tidak langsung melalui pheromonenya secara mendetail seperti status kerja, dll. Informasi yang ditukar memiliki ruang lingkup lokal, dimana hanya seekor semut yang terletak di tempat pheromone itu berada. Sistem ini disebut "Stigmergy" dan terjadi di banyak hewan yang hidup bersosial masyarakat (hal itu telah dipelajari dalam kasus pembangunan pilar dalam sarang rayap). Mekanisme untuk menyelesaikan masalah yang kompleks untuk ditangani oleh satu semut adalah contoh yang baik dari suatu sistem organisme. Sistem ini didasarkan pada feedback positif (menarik feromon semut lain yang akan memperkuat sendiri) dan negatif (disipasi dari rute oleh sistem mencegah penguapan dari labrakan). Secara teori, jika jumlah feromon tetap sama dari waktu ke waktu pada semua sisi, tidak ada rute yang akan dipilih. Namun, karena feedback, sedikit variasi pada sisi akan diperkuat dan dengan demikian memungkinkan pilihan sisi tersebut. Algoritma akan bergerak dari keadaan yang tidak stabil di mana tidak ada sisi yang lebih kuat daripada yang lain, untuk ke yang lebih stabil di mana jalur terdiri dari sisi paling kuat.

3. APLIKASI ANT COLONY OPTIMIZATION

Algoritma ant colony optimization telah digunakan dalam berbagai macam kasus optimasi kombinatorial, seperti persamaan kuadrat pada protein dan kendaraan bermotor dan banyak metoda yang diturunkan dari kasus dinamis dalam variabel riil, kasus stokastik, multi target, dan implementasi paralel.

Algoritma ini pun digunakan untuk memecahkan kasus yang sering kita dengar yaitu Travelling Salesman Problem (TSP). Algoritma ACO yang pertama disebut Ant system atau sistem semut dan ini ditujukan untuk memecahkan TSP, yang bertujuan untuk mencari jalur terpendek melingkar untuk mencapai berbagai kota. Algoritma umumnya relatif sederhana dan berdasarkan beberapa semut, masing –masing membuat kemungkinan perjalanan mengelilingi kota. Pada setiap grup, setiap semut memilih untuk pindah dari suatu kota ke kota lainnya dengan syarat:

1. Harus mengunjungi setiap kota tepat satu kali.
2. Kota yang lebih jauh memiliki kemungkinan dipilih yang lebih kecil dibanding yang lebih dekat.
3. Semakin kuat pheromone di setiap jalur, semakin kuat kemungkinan akan dipilih.
4. Setelah menyelesaikan perjalanan, semut meninggalkan pheromonenya di setiap jalur yang ia lewati jika jalurnya pendek.
5. Setelah beberapa pengulangan, jejak pheromone menguap.



Gambar 6. Travelling Salesman Problem

Contoh pseudo-code:

```
procedure ACO_MetaHeuristic
  while(not_termination)
    generateSolutions()
    daemonActions()
    pheromoneUpdate()
  end while
end procedure
```

Formula:

Pemilihan Sisi:

Seekor semut akan berjalan dari simpul i menuju simpul j dengan probabilitas

$$p_{i,j} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}$$

dimana

$\tau_{i,j}$ = jumlah pheromone pada sisi i,j

α = parameter pengontrol pengaruh $\tau_{i,j}$

$\eta_{i,j}$ = desirability sisi i,j (biasanya $1/d_{i,j}$, dimana d adalah jarak)

β = parameter pengontrol pengaruh $\eta_{i,j}$

Pheromone Update

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}$$

dimana

$\tau_{i,j}$ = jumlah pheromone pada sisi i,j

ρ = tingkat penguapan pheromone

and $\Delta\tau_{i,j}$ = jumlah pheromone dihasilkan

$$\Delta\tau_{i,j}^k = \begin{cases} 1/L_k & \text{Jika semut } k \text{ berjalan pada sisi } i,j \\ 0 & \text{lainnya} \end{cases}$$

dimana L_k adalah cost dari k th perjalanan semut (biasanya panjangnya)

4. KESIMPULAN

1. Ant Colony Optimization merupakan algoritma merupakan teknik probalistik untuk memecahkan masalah perhitungan dengan menemukan jalur terbaik melalui graf. Algoritma ini terinspirasi dari perilaku semut bersama dengan koloninya dalam mencari makanannya.
2. Banyak permasalahan yang membutuhkan optimasi yaitu suatu proses untuk mencapai hasil yang ideal atau optimal.
3. Ant Colony Optimization dapat digunakan dalam pemecahan masalah Travelling Salesman Problem (TSP).

REFERENSI

- [1] Munir, Rinaldi. (2006) "Matematika Diskrit". Bandung: Teknik Informatika ITB.
- [2] Wikipedia Indonesia. (2006). "Algoritma Semut" Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia." http://id.wikipedia.org/wiki/Algoritma_semut/. Waktu akses: 20 Desember 2009 pukul 21:00.
- [3] Wikipedia Indonesia. (2006). "Ant colony optimization" Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Ant_colony_optimization/. Waktu akses: 20 Desember 2006 pukul 21:00.
- [4] Wikipedia Indonesia. (2006). "Optimisasi" Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia." <http://id.wikipedia.org/wiki/Optimisasi/>. Waktu akses: 20 Desember 2009 pukul 22:00.